**HTML**

学习视频：<https://www.bilibili.com/video/BV1sW411T78k?from=search&seid=14839709518398153089>

**一、学习内容简介**

1.html：结构、css：表现、JavaScript：行为。

**二、HTML简介**

1.HTML：超文本标记语言。

2.主要使用标签进行标记，一般是成对出现。

3.网页的标准格式：

|  |
| --- |
| <html> *-----根标签*  <head> *-----头部信息，帮助浏览器解析网页*  <title></title> *-----网页标题*  </head>  <body></body> *-----页面主体内容*  </html> |

**三、HTML注释**

1.<!—注释内容 -->

2.搜索引擎在检索页面时，首先会检索<title>标签。

3.注释习惯：

|  |
| --- |
| *<!--*  *功能：*  *作者：*  *日期：*  *-->* |

**四、标签的属性**

1.可以在开始标签，为标签添加属性。属性名=”属性值”。

html参考手册：<https://www.w3school.com.cn/tags/index.asp>

2.关于样式的属性，不建议使用，而建议采用css的方式来进行设置。

**五、文档的声明**

1.告知浏览器网页的版本：

|  |
| --- |
| *<!-- HTML5版本 -->*  <!DOCTYPE html> |

**六、乱码的问题**

1.原因：

编码与解码采用的字符集不同。

2.字符集：

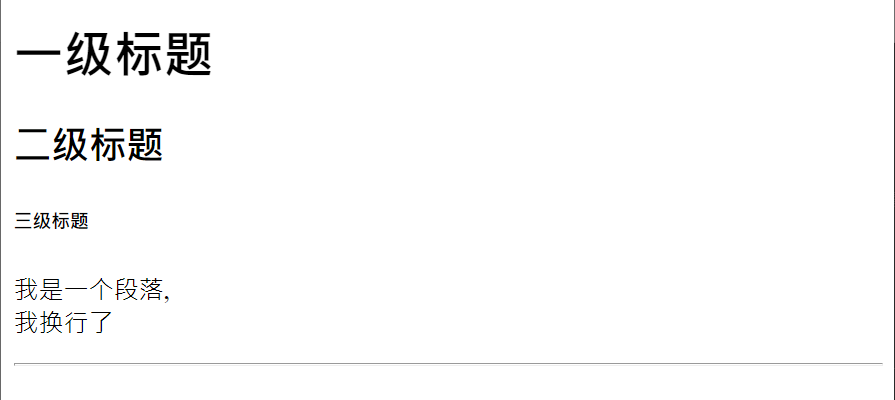
ASCII：7bit、ISO-8859-1：8bit、GBK、**UTF-8：万国码**

3.中文系统中的浏览器中，默认使用GBK2312解码。

|  |
| --- |
| <meta charset=”UTF-8”/> |

**七、常用标签**

|  |
| --- |
| <!DOCTYPE html> <html>  <head>  <meta charset="utf-8"/>  <title>常用的标签</title>  </head>  <body>  <!--  标题标签  h1~h6，注意标题的大小不重要，重要的是看标题的级别  -->  <h1>一级标题</h1>  <h2>二级标题</h2>  <h6>三级标题</h6>  <!--  段落标签  换行与空格一律是解析成一个空格  换行标签  -->  <p>我是一个段落,<br/>  我换行了  </p>  <!--  水平线标签  -->  <hr/>  </body> </html> |



**八、实体（转义字符）**

1.某些特殊字符会被浏览器解析，如<>，但实际上是不希望被解析为标签的，因此需要使用转义字符。

2.&实体名称;

|  |  |
| --- | --- |
| 转义字符 | 含义 |
| &lt; | < |
| &gt; | > |
| &nbsp; | 空格 |
| &copy; | 版权符号 |

**九、图片标签**

|  |
| --- |
| <!DOCTYPE html> <html lang="en"> <head>  <meta charset="UTF-8">  <title>图片标签</title> </head> <body>  <!--  图片标签  src：图片的路径  alt：图片描述，在图片不能显示时出现；搜索引擎可以通过alt来识别图片  width、height：当只设置宽或者高时，则另一个属性自动同比例缩放  -->  <img src="./static/img/1.jpg" width="500px"/> </body> </html> |



**十、相对路径**

1.相对路径：相对于当前资源所在位置。

2.返回上一级：../

十一、图片的格式

**1.几种图片的格式：**

|  |  |  |
| --- | --- | --- |
| 图片格式 | 特点 | 应用 |
| JPEG | 1.支持的颜色比较多  2.图片可以压缩  3.不支持透明 | 保存照片等颜色丰富的图片 |
| GIF | 1.支持颜色比较少  2.支持简单的透明  3.支持动态图 | 图片颜色单一，或者动态图 |
| PNG | 1.支持颜色多  2.支持复杂的透明 | 显示颜色复杂的透明的图片 |

2.图片的使用原则：

（1）效果不一致，使用好的；

（2）效果一致，使用小的；

**十二、meta**

|  |
| --- |
| <html>  <head>  <title>chocho的第一个网页</title>  <!--网页解码方式-->  <meta charset="utf-8"/>  <!--  网页关键字  name="keywords"：为固定格式  多个关键字可以使用,分割  -->  <meta name="keywords" content="HTML5, JavaScript, 前端"/>  <!--  网页描述  搜索引擎在检索页面时，会同时检索关键字和描述  -->  <meta name="description" content="这是一个html5网页"/>  <!--  做请求的重定向  http-equiv="refresh"：表示刷新  content：分号前面表示几秒后开始重定向，分号后面表示重定向的url  -->  <meta http-equiv="refresh" content="5;url=http://www.baidu.com" />  </head>  <body>  chocho的第一个网页  </body> </html> |

**十三、语法规范**

1.不区分大小写，但是一般使用小写

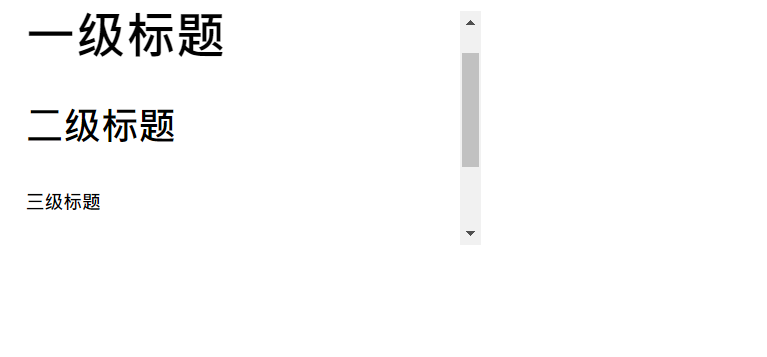
2.注释不能嵌套

3.值必须加引号

**十四、内联框架**

1.引入外部页面

|  |
| --- |
| <!DOCTYPE html> <html lang="en"> <head>  <meta charset="UTF-8">  <title>内联框架</title> </head> <body>  <!--不推荐使用-->  <iframe src="demo05.html" frameborder="0"></iframe> </body> </html> |



**十五、超链接**

|  |
| --- |
| <!DOCTYPE html> <html lang="en"> <head>  <meta charset="UTF-8">  <title>超链接</title> </head> <body>  <!--  超链接  href：连接的目标地址  target：超链接打开的位置。  \_self：在当前窗口打开，默认值；  \_blank：在新的窗口打开；  内联框架名称：在内联框架内显示；  -->  <a href="http://www.baidu.com">我是超链接</a> </body> </html> |

1.如何去模仿一个网站？

（1）从上到下，从外到内

（2）先模仿内容，选择合适的标签

（3）调整位置、样式

（4）不断比对效果，逐渐调整或者查看源码，直到模仿完成

2.锚点的使用

|  |
| --- |
| <!--语法：#锚点id名--> <a href="#bottom">底部</a>  <!--id设置锚点，在页面中必须唯一--> <a href="#" id="bottom">回到顶部</a> | <a href="#">联系我们</a> |

3.超链接到邮箱

|  |
| --- |
| <a href="mailto:chocho@qq.com">联系我们</a> |

点击后自动启动系统默认的邮箱系统。

**十六、列表**

1.ul无序列表，li表示列表项

（1）type属性可以指定项目符号，不推荐使用

|  |  |
| --- | --- |
| 属性值 | 效果 |
| disc | 默认值，实心原点 |
| square | 实心方块 |
| circle | 空心圆 |

（2）去除项目符号

|  |
| --- |
| ul{  list-style: none;  } |

（3）应用

（1）新闻列表

（2）导航栏

2.ol有序列表

（1）type属性可以指定序号类型

|  |  |
| --- | --- |
| 属性值 | 效果 |
| 1 | 默认值，数字 |
| a、A | 英文字母 |
| I | 罗马数字 |

3.定义列表

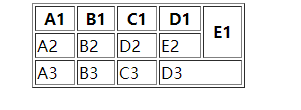
用于对一些词汇或者内容进行定义：

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <meta charset="utf-8">  <title>定义列表</title>  </head>  <body>  <!--  dl：定义一个定义列表  dt：被定义内容  dd：对被定义内容进行描述  -->  <dl>  <dt>武松</dt>  <dd>景阳冈打虎英雄，战斗力99</dd>  <dd>后打死西门庆，投奔梁山</dd>    <dt>武大</dt>  <dd>著名餐饮企业家，战斗力0</dd>  </dl>  </body>  </html> |

可以应用于下拉列表。

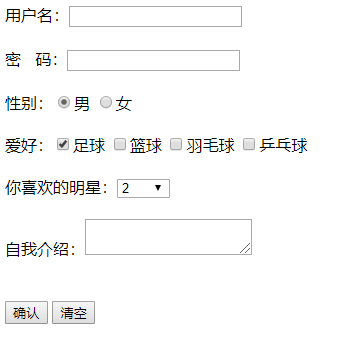
**十七、表格**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <meta charset="utf-8">  <title>表格</title>  </head>  <body>  <table border="1" width="40%" align="center">  <tr>  <th>A1</th>  <th>B1</th>  <th>C1</th>  <th>D1</th>  <!-- 纵向合并 -->  <th rowspan="2">E1</th>  </tr>  <tr>  <td>A2</td>  <td>B2</td>  <td>D2</td>  <td>E2</td>  </tr>  <tr>  <td>A3</td>  <td>B3</td>  <td>C3</td>  <!-- 横向合并 -->  <td colspan="2">D3</td>  </tr>  </table>  </body>  </html> |



**十八、表单**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <meta charset="utf-8">  <title>表单</title>  </head>  <body>  <!-- action：为表单提交到服务器的地址 -->  <form action="#">  <!-- 文本框 -->  用户名：<input type="text" name="username"/><br/><br/>    <!-- 密码框 -->  密&nbsp;&nbsp;&nbsp;码：<input type="password" name="password"/><br/><br/>    <!-- 单选按钮  name相同的按钮是一组  -->  性别：<input type="radio" name="gender" value="male" checked="checked"/>男  <input type="radio" name="gender" value="female"/>女<br/><br/>    <!-- 多选框 -->  爱好：<input type="checkbox" name="hobby" value="zq" checked="checked">足球  <input type="checkbox" name="hobby" value="lq">篮球  <input type="checkbox" name="hobby" value="ymq">羽毛球  <input type="checkbox" name="hobby" value="ppq">乒乓球<br/><br/>    <!-- 下拉列表  name：指定给下拉列表  value：指定给列表项  -->  你喜欢的明星：<select name="star">  <optgroup label="数字">  <option value ="1">1</option>  <!-- 设置默认值 -->  <option value ="2" selected="selected">2</option>  <option value ="3">3</option>  <option value ="4">4</option>  <option value ="5">5</option>  </optgroup>  <optgroup label="字母">  <option value ="A">A</option>  <option value ="B">B</option>  <option value ="C">C</option>  </optgroup>  </select><br/><br/>    <!-- 文本域 -->  <!-- label标签 -->  <label for="selfintro">自我介绍：</label><textarea id="selfintro" info="selfintro">  </textarea><br/><br/><br/>    <!-- 提交按钮，可以将表单中的信息全部提交 -->  <input type="submit" value="确认"/>  <!-- 重置按钮 -->  <input type="reset" value="清空">  </form>  </body>  </html> |



**CSS**

**一、css简介**

1.css为层叠样式表，即样式一层一层地叠加产生的效果。用户看到的是最顶层的效果。

工具tilt可以看到层叠效果。

2.css设置方式

（1）写在元素的style属性中：

|  |
| --- |
| <p style="color:red;font-size: 20px;">  锄禾日当午，汗滴禾下土</p> |

又称内联样式，只对当前标签起作用。当样式出现大量耦合时，则不方便复用。



（2）写在head的<style>标签中：

|  |
| --- |
| <!--type="text/css"为默认值，但一些旧版本浏览器不兼容，因此建议写上--> <style type="text/css">  p{  color: red;  font-size: 20px;  } </style> |

便于页内的样式复用，但不方便页面间的样式复用。

（3）写在css文件中：

|  |
| --- |
| style.css |
| p{  color: red;  font-size: 20px; } |

引入css文件：

|  |
| --- |
| demo06.html |
| <link rel="stylesheet" type="text/css" href="./static/css/style.css"> |

可以最大限度使得结构与样式分离，且利用浏览器缓存，加快页面访问速度。

**二、css语法**

1.css的注释：/\*\*/

2.选择器 {声明1；声明2……}

工具Hbuilder

**三、内联与块元素**

（1）块元素：会独占一行的元素。

div、p、br、h1

\*div没有设置任何默认样式，div主要用于页面布局。

（2）内联元素：行内元素，只占自身大小。

span、a、img、iframe

\*span没有设置任何默认样式，用于选中文字，便于为其设置样式。

（3）原则：

一般块元素包含行内元素；

a元素可以包含任意元素；

p元素中不能放任何块元素；

**四、常用选择器**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <meta charset="utf-8">  <title>常用选择器</title>  <style type="text/css">  /\*  范围：页面所有元素  样式：  字体：红色  \*/  /\*元素选择器\*/  p{  color: red;  }    /\*  范围：指定p元素  样式：  字体大小：20  \*/  /\* id选择器\*/  #p1{  font-size: 20px;  }    /\* 类选择器 \*/  .p{  color: greenyellow;  }    /\* 选中所有元素 \*/  \*{  opacity: 0.8;  }    /\* 复合选择器 \*/  span.p2{  font-size: 30px;  }      </style>  </head>  <body>  <h1>chocho</h1>  <p>秋秋的前端之旅</p>  <p id="p1">秋秋的前端之旅</p>  <p class="p">秋秋的前端之旅</p>  <p class="p">秋秋的前端之旅</p>  <p class="p">秋秋的前端之旅</p>  <p class="p">秋秋的前端之旅</p>  <p>秋秋的前端之旅</p>  <p>秋秋的前端之旅</p>  <p><span class="p2">秋秋</span>的前端之旅</p>  <p>秋秋的前端之旅</p>  </body>  </html> |



**五、子代选择器**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <meta charset="utf-8">  <title>子代选择器</title>  <style type="text/css">  /\* 同一个祖先的后代元素 \*/  div span{  color: #ADFF2F;  }    /\* 统一祖先的后代的后代元素 \*/  div p span{  color: #FF0000;  }    /\* 父元素 > 子元素 \*/  div > span{  font-size: 30px;  }  </style>  </head>  <body>  <!-- 祖先元素：div相对于span来说是祖先元素  父元素：直接包含子代元素  子元素：直接呗父元素包含  -->  <div>  <p>  <span>我是span</span>  </p>  <span>我是div中的span</span>  </div>  <span>我是body中的span</span>  </body>  </html> |



css练习：<https://github.com/yizhou19/select_css_practice>

**六、伪类选择器**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <meta charset="utf-8">  <title>伪类选择器</title>  <style type="text/css">  /\* 伪类：用于表示元素的一种特殊状态\*/    /\*  范围：没访问过的链接  样式：  绿色  \*/  a:link{  background-color: #ADFF2F;  }    /\*  范围：访问过的链接  样式：  红色    浏览器通过历史记录来判断是否访问过，由于涉及隐私，只能设置字体颜色!!!  \*/  a:visited{  color: #FF0000;  }    /\* 范围：鼠标悬浮其上  样式：  透明度：0.8  \*/  a:hover{  opacity: 0.8;  }    /\* 范围：超链接被点击的状态  样式：  蓝色  \*/  a:active{  color: blue;  }    /\* 范围：文本框被选中，获取焦点  样式：  背景黄色  \*/  input:focus{  background-color: #ADFF2F;  }    /\* 范围：选中的文字  样式：  aqua  \*/  p::selection{  color: aqua;  }  /\*火狐  p::-moz-selection{  color: aqua;  } \*/  </style>  </head>  <body>  <a href="http://www.baidu.com">没有被访问过的链接</a><br/>  <a href="http://www.baidu123.com">已经被访问过的链接</a><br/>    <input type="text"/>    <p>我是一段文字</p>  </body>  </html> |

**七、伪元素**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <meta charset="utf-8">  <title>伪元素</title>  <style type="text/css">  /\* 范围：第一个字母  样式：  大小30  \*/  p::first-letter{  font-size: 30px;  }    /\* 范围：第一行，自适应  样式：  黄色  \*/  p::first-line{  color: #ADFF2F;  }    /\* 范围：元素最前面、元素最后面  样式：  需要结合content使用\*/  p:before{  content: '最前面<';  }  p:after{  content: '>最后面';  }  </style>  </head>  <body>  <p>我是一个段落  我是填充字段我是填充字段我是填充字段我是填充字段我是填充字段我是填充字段  我是填充字段我是填充字段我是填充字段我是填充字段我是填充字段</p>  </body>  </html> |

**八、属性选择器**

1.title属性：当鼠标悬浮，则显示提示文字。

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <meta charset="utf-8">  <title>属性选择器</title>  <style type="text/css">  /\* 范围：具有相同属性的元素  样式：  黄色背景  \*/  p[title]{  background-color: #ADFF2F;  }    /\* 范围：具有指定属性的元素  样式：  红色  \*/  p[title='hh1']{  color: #FF0000;  }    /\* 范围：以指定开头开头的属性  样式：  蓝色  \*/  p[title^='ab']{  color: blue;  }    /\* 范围：以指定结尾结尾的属性  样式：  白色  \*/  p[title$='2']{  color: white;  }    /\* 范围：包含指定字符串的属性  样式：  背景颜色橙色  \*/  p[title\*='k']{  background-color: orange;  }  </style>  </head>  <body>  <p title="hh1">hha</p>  <p title="hh2">hha</p>  <p title="abc">hha</p>  <p title="kgffff">hha</p>  <p title="dsdk">hha</p>  <p>hha</p>  </body>  </html> |

**九、子元素的伪类**

|  |  |
| --- | --- |
| 名称 | 说明 |
| :first-child | 所有子标签中的第一个子标签 |
| :last-child | 所有子标签中的最后一个子标签 |
| :nth-child(n) | 所有子标签中的第n个子标签，even表示偶数，odd表示奇数 |
| :first-of-type | 当前类型中的第一个子标签 |
| :last-of-type | 当前类型中的最后一个子标签 |
| :nth-of-type | 当前类型中的第n个子标签 |

**十、兄弟元素选择器**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <meta charset="utf-8">  <title>兄弟元素选择器</title>  <style type="text/css">  /\* 范围：第一个兄弟元素  样式：  红色    兄弟元素必须是紧挨着的  \*/  span + p{  color: red;  }    /\* 范围：后面所有兄弟元素  样式：  黄色背景  \*/  span ~ p{  background-color: #00FFFF  }  </style>  </head>  <body>  <p>这是一个段落</p>  <span>这是一个span</span>  <p>这是一个段落</p>  <p>这是一个段落</p>  </body>  </html> |

**十一、否定伪类**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <meta charset="utf-8">  <title>否定伪类</title>  <style type="text/css">  /\* 范围：所有p元素，除了class='p' 元素  格式：:not(选择器)\*/  p:not(.p){  color: red;  }  </style>  </head>  <body>  <p class="p">这是一个段落</p>  <p>这是一个段落</p>  <p>这是一个段落</p>  </body>  </html> |

**十二、样式的继承**

1.子元素会继承父元素的样式。

2.并不是所有的样式都会被继承，比如与背景相关的。

**十三、选择器的优先级**

1.当样式发生冲突（不同选择器，设置同一个样式）时，最终采用的选择器是优先级最高的。

|  |  |
| --- | --- |
| 选择器名称 | 优先级 |
| 内联样式 | 1000 |
| id选择器 | 100 |
| 类和伪类 | 10 |
| 元素选择器 | 1 |
| 通配\* | 0 |
| 继承的样式 | 没有优先级 |

2.当包含多种选择器，则需要将各个选择器的优先级相加。

3.若优先级都一样，则选择靠后的选择器。

4.important!可以将优先级调整到最高。

**十四、长度单位**

|  |  |
| --- | --- |
| 长度单位 | 说明 |
| px | 像素 |
| 百分比% | 根据其父元素进行计算，可以自适应 |
| em | 相对于当前元素字体大小计算，1em=1font-size |

**十五、十六进制RGB值**

红色：red、蓝色：blue、绿色：green

rgb(红色，绿色，蓝色)

1.颜色浓度

（1）0~255，0表示没有，255表示很大。

（2）0%~100%，0%表示0,100%表示255。

（3）十六进制，#红色绿色蓝色，每种颜色范围00~FF，00表示0，FF表示255。

**十六、字体样式**

|  |  |
| --- | --- |
| 字体样式 | 说明 |
| color | 字体颜色 |
| font-size | 字体大小，默认大小是16px |
| font-family | 字体，浏览器使用的是计算机中的字体，font-family: ‘curlz mt’ |
| font-style | 字体样式，italic斜体、oblique倾斜 |
| font-weight | 文本加粗效果，bold加粗、lighter更细 |
| font-variant | small-caps小型大写字母 |
| font | 将以上属性值按空格分隔，没有顺序要求；大小、字体必须写；字体必须最后 |

**十七、行高**

line-height：行高，可以使用数值或者百分比去表示。

**十八、文本样式**

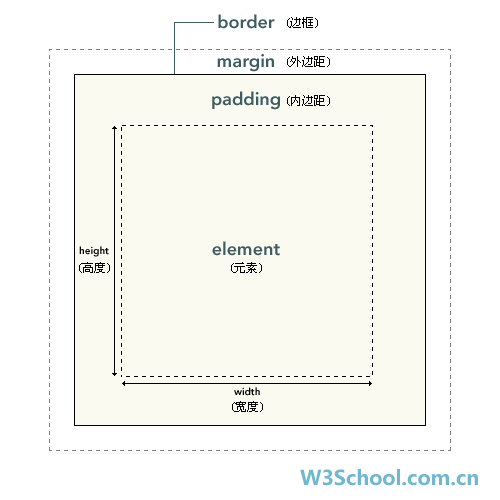
|  |  |
| --- | --- |
| 文本样式 | 说明 |
| text-decoration | underline下划线、overline上划线、line-through删除线、none去掉样式 |
| text-align | center居中、left左对齐、right右对齐、justify两端对齐 |
| text-indent | 首行缩进，2em |
| font-style | 字体样式，italic斜体、oblique倾斜 |
| font-weight | 文本加粗效果，bold加粗、lighter更细 |
| font-variant | small-caps小型大写字母 |
| font | 将以上属性值按空格分隔，没有顺序要求；大小、字体必须写；字体必须最后 |

**十九、盒子模型**

1.作用

css在处理网页时，每个元素都包含在一个不可见的盒子里；那么对于网页的布局就是摆放盒子；

2.特点



（1）具有边框boder

（2）边框与内容之间是内边距padding

（3）边框与边框之间是外边距margin

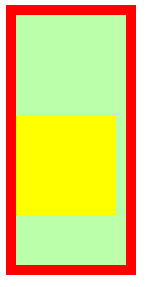
**二十、盒子模型——边框**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <meta charset="utf-8">  <title>盒子模型——边框</title>  <style type="text/css">  .box1{  /\* 设置内容区的高度、宽度 \*/  width: 100px;  height: 100px;  /\* 背景颜色 \*/  background-color: #bfa;  /\* 边框宽度、颜色、样式 \*/  /\* border-width: 10px; \*/  /\* 分别设置边框宽度 上 右 下 左-->顺时针方向\*/  border-width: 10px 20px 30px 40px;  border-color: yellowgreen;  /\*solid实线 dotted点状 dashed虚线 double双线 \*/  border-style: solid;    /\* boder属性可以直接设置：样式、宽度、颜色 \*/  }  </style>  </head>  <body>  <div class="box1"></div>  </body>  </html> |



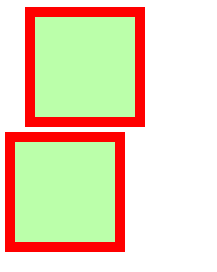
**二十一、盒子模型——内边距**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <meta charset="utf-8">  <title>盒子模型——内边距</title>  <style type="text/css">  .box1{  width: 100px;  height: 100px;  /\* 元素背景会延升到内边距 \*/  background-color: #bfa;  border: 10px red solid;  /\* 内边距 盒子内容与边框之间的距离\*/  padding-top: 100px;  padding-right: 10px;  padding-bottom: 50px;  /\* 盒子大小由边框+内容区+内边距决定 \*/  }    .box2{  width: 100%;  height: 100%;  background-color: yellow;  }  </style>  </head>  <body>  <div class="box1">  <div class="box2"></div>  </div>  </body>  </html> |



**二十二、盒子模型——外边距**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <meta charset="utf-8">  <title>盒子模型——外边距</title>  <style type="text/css">  .box1{  height: 100px;  width: 100px;  background-color: #bfa;  border: 10px solid red;  /\* 外边距  一般给左右外边距设置auto\*/  margin-top: 10px;  margin-left: 20px;  margin-bottom: 5px;  }    .box2{  height: 100px;  width: 100px;  background-color: #bfa;  border: 10px solid red;  }  </style>  </head>  <body>  <div class="box1">    </div>  <div class="box2"></div>  </body>  </html> |



**二十三、垂直外边距的重叠**

1.当两个垂直相邻盒子模型，垂直外边距重复，则会发生重叠，取最大值。

当两个嵌套的盒子模型，此时两个盒子垂直相邻，因此在0和最大值之间取最大值。

2.为了消除这个垂直边距重叠的效果，则为盒子设置边框宽度、设置内边距，则可以把两个盒子隔开；或者利用垂直边距重叠的效果，进行挤压，完成布局。

**二十四、浏览器默认样式**

1.body等标签都有默认的边距。

2.清除浏览器默认样式

|  |
| --- |
| \*{  margin: 0;  padding: 0;  } |

**二十五、内联元素的盒模型**

1.内联元素不能设置width、height。

2.内联元素可以设置水平边距/边框，可以设置垂直边距/边框，但不会影响布局。

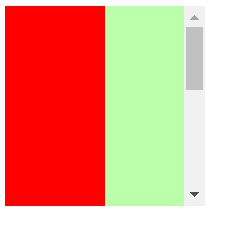
3.水平方向外边距不会重叠，会求和，不支持垂直外边距。

**二十六、display和visibility**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <meta charset="utf-8">  <title>display</title>  <style type="text/css">  a{  background-color: #bfa;  width: 500px;  height: 500px;  /\* 将内联元素变成块元素  block：块元素  inline：内联元素  inline-block：行内块元素  none：隐藏元素，其元素不继续占有空间  \*/  display: block;  /\* 规定元素是否可见  visible：可见的  hidden：不可见的，但是元素仍然占据布局  \*/  }  </style>  </head>  <body>  <a href="#">我是一个大大的超链接</a>  </body>  </html> |

**二十七、overflow**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <meta charset="utf-8">  <title>overflow</title>  <style type="text/css">  .box1{  width: 200px;  height: 200px;  background-color: #bfa;  /\* 溢出  visible：默认值，显示溢出部分  hidden：会修剪溢出部分  scroll：溢出部分可以使用滚动条查看，为父元素添加滚动条  auto：根据需要添加水平、垂直滚动条  \*/  overflow: auto;  }    .box2{  width: 100px;  height: 500px;  background-color: red;  }  </style>  </head>  <body>  <div class="box1">  <div class="box2"></div>  </div>  </body>  </html> |



**二十八、文档流**

1.文档流处于网页最底层，它表示的是一个页面中的位置。我们所创建的元素默认在文档流中。

2.元素在文档流中的特点

（1）块元素

\*独占一行，自上向下排列

\*默认宽度是父元素的100%

\*默认高度被子元素撑开

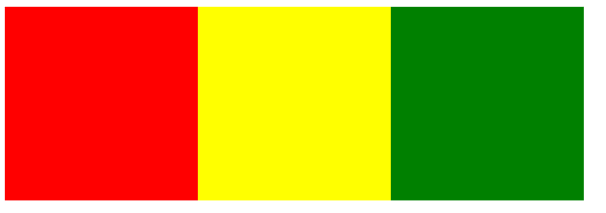
（2）内联元素

\*只占自身大小，自左向右排列，若一行不足以容纳，则换到下一行

\*默认高度宽度被子元素撑开

**二十九、浮动**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <meta charset="utf-8">  <title>浮动</title>  <style type="text/css">  .box1{  width: 200px;  height: 200px;  background-color: red;  /\* 在文档流中垂直排列，如果希望水平排列，则脱离文档流 \*/  /\* float：  none：默认值，不浮动，在文档流中  left：脱离文档流，左侧浮动  right：脱离文档流，右侧  \*/  /\* 左浮动，则文档流剩余元素依旧按照文档流规则显示  尽量向文档左上、右上漂浮\*/  float: left;  }    .box2{  width: 200px;  height: 200px;  background-color: yellow;  float: left;  }    .box3{  width: 200px;  height: 200px;  background-color: green;  float: left;  }  </style>  </head>  <body>  <div class="box1"></div>  <div class="box2"></div>  <div class="box3"></div>  </body>  </html> |



（1）元素浮动以后，会尽量往左上或者右上浮动

（2）直到遇到父元素边框，或者其他浮动元素

（3）若浮动元素上边是一个没有浮动的块元素，则浮动元素不会超过块元素

（4）浮动元素不会超过兄弟元素，最多与之对齐

（5）浮动的元素不会盖住文字，文字会自动环绕在浮动元素周围——可以实现文字环绕图片效果



（6）内联元素脱离文档流，会变成块元素

**三十、简单的布局**

1.布局分类

（1）固定布局

（2）自适应布局（根据设备不同，布局显示会变化）

2.看懂网页结构

其实就是看懂网页的盒子布局

3.模仿编写一个简单的网页

（1）设计头部、主体、底部的布局

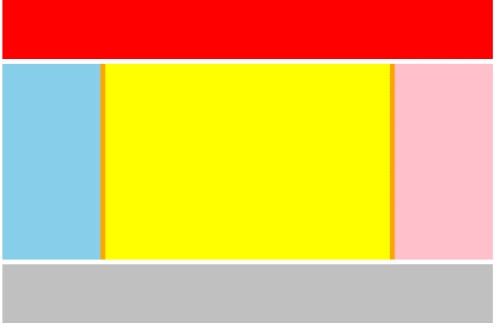
（2）添加样式：宽高、背景、边距、位置 ；注意父子及位置关系

（3）取消默认样式

（4）继续对每个布局进行迭代，（1）（2）步

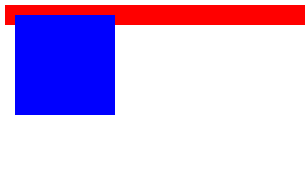
4.一个简单的页面布局（固定布局）

|  |
| --- |
| **简单的布局.html** |
| <!DOCTYPE html>  <html>  <head>  <meta charset="utf-8">  <title>简单的布局</title>  <link rel="stylesheet" type="text/css" href="static/css/chocho.css"/>  </head>  <body>  <!-- 头部 -->  <div class="header"></div>    <!-- 主体内容 -->  <div class="content">    <!-- 左侧内容 -->  <div class="left"></div>    <!-- 中间内容 -->  <div class="center">    <!-- 中间-上面部分 -->  <div class="center-top"></div>    <!-- 中间-下面部分 -->  <div class="center-bottom"></div>  </div>    <!-- 右侧内容 -->  <div class="right"></div>  </div>    <!-- 底部信息 -->  <div class="footer"></div>  </body>  </html> |
| **chocho.css** |
| /\* 去除默认样式 \*/  \*{  margin: 0;  padding: 0;  }  /\* 设置头部div \*/  .header{  width: 1000px;  height: 120px;  background-color: red;  margin: 0 auto;  }  /\* 设置主体内容 \*/  .content{  width: 1000px;  height: 400px;  background-color: orange;  margin: 10px auto;  }  /\* 左侧内容 \*/  .left{  width: 200px;  height: 100%;  background-color: skyblue;  float: left;  }  /\* 中间内容 \*/  .center{  width: 580px;  height: 100%;  background-color: yellow;  float: left;  margin: 0 10px;  }  /\* 右侧内容 \*/  .right{  width: 200px;  height: 100%;  background-color: pink;  float: left;  }  /\* 设置底部 \*/  .footer{  width: 1000px;  height: 120px;  background-color: silver;  margin: 0 auto;  } |



**三十一、高度塌陷问题**

1.什么是高度塌陷



当为子元素设置浮动后，子元素会完全脱离文档流。此时父元素没有内容，则高度塌陷。最终可能会影响页面布局。在开发中要避免出现高度塌陷的问题。

2.解决方案

（1）为父元素设置高度，但不能自适应

（2）开启父元素的BFC

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <meta charset="utf-8">  <title>解决高度塌陷1</title>  <style type="text/css">  .box1{  border: 10px red solid;  zoom: 1;  overflow: hidden;  }    .box2{  width: 100px;  height: 100px;  background-color: yellow;  float: left;  }  </style>  </head>  <body>  <div class="box1">  <div class="box2"></div>  </div>  </body>  </html> |



（3）最终解决方案！！！

clear可以清除其他元素浮动对当前元素的影响

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <meta charset="utf-8">  <title>高度坍塌最终解决方案</title>  <style type="text/css">  .box1{  width: 100px;  height: 100px;  background-color: #bfa;  /\* float: left; \*/  }    .box11{  width: 50px;  height: 50px;  background-color: pink;  float: left;  }    /\*解决方案二 \*/  .clear{  clear: both;  }    **/\* 解决方案三 \*/**  **.clearfix:after{**  **content: '';**  **display: block;**  **clear: both;**  **}**    .box2{  width: 200px;  height: 200px;  background-color: yellow;  /\* 清除浮动影响  clear:  none:默认值，不清楚浮动  left：去除左侧元素浮动影响  right:去除右侧元素浮动影响  both:清除两侧影响  清除浮动后，元素回到浮动前的位置    解决方案一  \*/  /\* clear: left; \*/  }    .box3{  width: 300px;  height: 300px;  background-color: pink;  }  </style>  </head>  <body>  <div class="box1 clearfix">  <div class="box11"></div>  <div class="clear"></div>  </div>  <div class="box2"></div>  <div class="box3"></div>  </body>  </html> |

**三十二、导航条**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <meta charset="utf-8">  <title>导航条</title>  <style type="text/css">  /\* 去除默认样式 \*/  \*{  margin: 0;  padding: 0;  }    /\* 设置ul \*/  ul{  /\* 去除项目符号 \*/  list-style: none;  /\* 添加背景颜色 \*/  background-color: skyblue;  width: 1000px;  /\* 元素居中 \*/  margin: 50px auto;  /\* 解决高度塌陷 \*/  zoom: 1;  overflow: hidden;  }    /\* 当鼠标移入时，可以背景变成红色 \*/  .nav a:hover{  background-color: #FF0000;  }    /\* 设置li \*/  li{  float: left;  }    /\* 设置a \*/  li a{  /\* 转化为块元素 \*/  display: inline-block;  width: 250px;  text-align: center;  padding: 5px 0;  /\* 去除下划线 \*/  text-decoration: none;  color: white;  font-weight: bold;    }  </style>  </head>  <body>  <ul class="nav">  <li><a href="#">首页</a></li>  <li><a href="#">新闻</a></li>  <li><a href="#">联系</a></li>  <li><a href="#">关于</a></li>  </ul>  </body>  </html> |



**三十三、开班信息**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <meta charset="utf-8">  <title>开班信息</title>  <style type="text/css">  /\* 清除默认样式 \*/  \*{  margin: 0;  padding: 0;  }    /\* 统一页面字体 \*/  body{  font: 10px 宋体;  }    /\* 设置outer的大小、位置 \*/  .outer{  width: 300px;  /\* height: 473px; \*/  /\* background-color: #bfa; \*/  margin: 50px auto;  }    /\* 设置title的边框 \*/  .title{  border-top: 2px #019e8b solid;  height: 36px;  background-color: #f5f5f5;  /\* 垂直居中 \*/  line-height: 36px;  padding: 0 22px 0 16px;  }    /\* 设置title中的标题浮动 \*/  .title h3{  float: left;  /\* 默认值将行高覆盖了 \*/  font: 16px/36px '微软雅黑';  }    /\* 设置title中的超链接浮动 \*/  .title a{  float: right;  color: red;  }    /\* 设置内容 \*/  .content{  border: 1px solid #deddd9;  padding: 0px 28px 0px 22px;  }    /\* 设置内容中的标题 \*/  .content h3{  margin-top: 14px;  margin-bottom: 16px;  }    /\* 设置内容超链接 \*/  .content a{  color: black;  text-decoration: none;  font-size: 12px;  }    /\* 设置超链接hover的伪类 \*/  .content a:hover{  color: red;  }    /\* 设置ul \*/  .content ul{  list-style: none;  /\* 为ul设置下边框 \*/  border-bottom: 1px dashed #deddd9;  }    /\* 取消下边框 \*/  .content .no-border{  border: none;  }    /\* 设置内容中的li \*/  .content ul li{  margin-bottom: 14px;  }    /\* 设置内容中的红色字体 \*/  .content .red{  color: red;  font-weight: bold;  }    /\* 右浮动 \*/  .content .right{  float: right;  }  </style>  </head>  <body>  <!-- 外层div，容纳整个内容 -->  <div class="outer">    <!-- 开班信息的标题 -->  <div class="title">  <!-- 标题 -->  <h3>近期开班</h3>  <a href="#">16年面授开班计划</a>  </div>    <!-- 开班信息的主体内容 -->  <div class="content">  <!-- 1 -->  <!-- 标题 -->  <h3><a href="#">JavaEE+云计算-全程就业班</a></h3>  <!-- 开班信息 -->  <ul>  <li>  <a href="#">开班时间：<span class="red">2016-04-27</span></a>  <a class="right" href="#"><span class="red">预约报名</span></a>  </li>  <li>  <a href="#">开班时间：<span class="red">2016-04-07</span></a>  <a class="right" href="#"><span class="red">无座，名额爆满</span></a>  </li>  <li>  <a href="#">开班时间：<span>2016-03-15</span></a>  <a class="right" href="#"><span>开班盛况</span></a>  </li>  <li>  <a href="#">开班时间：<span>2016-02-25</span></a>  <a class="right" href="#"><span>开班盛况</span></a>  </li>  <li>  <a href="#">开班时间：<span>2015-12-26</span></a>  <a class="right" href="#"><span>开班盛况</span></a>  </li>  </ul>    <!-- 2 -->  <!-- 标题 -->  <h3><a href="#">Andriod+人工智能-全程就业班</a></h3>  <!-- 开班信息 -->  <ul>  <li>  <a href="#">开班时间：<span class="red">2016-04-10</span></a>  <a class="right" href="#"><span class="red">预约报名</span></a>  </li>  <li>  <a href="#">开班时间：<span>2016-03-17</span></a>  <a class="right" href="#"><span>开班盛况</span></a>  </li>  <li>  <a href="#">开班时间：<span>2016-02-20</span></a>  <a class="right" href="#"><span>开班盛况</span></a>  </li>  <li>  <a href="#">开班时间：<span>2015-12-23</span></a>  <a class="right" href="#"><span>开班盛况</span></a>  </li>  </ul>    <!-- 3 -->  <!-- 标题 -->  <h3><a href="#">前端+HTML5-全程就业班</a></h3>  <!-- 开班信息 -->  <ul class="no-border">  <li>  <a href="#">开班时间：<span class="red">2016-05-10</span></a>  <a class="right" href="#"><span class="red">预约报名</span></a>  </li>  <li>  <a href="#">开班时间：<span>2016-03-16</span></a>  <a class="right" href="#"><span>开班盛况</span></a>  </li>  </ul>  </div>  </div>  </body>  </html> |



**三十四、定位**

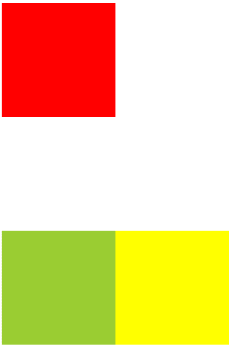
1.position

|  |  |
| --- | --- |
| 属性值 | 说明 |
| static | 默认值，没有开启定位 |
| relative | 元素相对定位，相对于定位位置；不会脱离文档流 |
| absolute | 绝对定位，脱离文档流 |
| fixed | 固定定位，脱离文档流 |

2.top、bottom、left、right

（1）相对定位（相对于原来所在位置）

|  |
| --- |
| .box2{  width: 200px;  height: 200px;  background-color: yellow;  /\* 相对定位，相对定位使得元素提升一个层级，可以覆盖文档流的内容  相对定位对内联元素不会有影响，不能设置宽高  \*/  position: relative;  left: 200px;  top: 200px;  } |



（2）绝对定位（相对于离他最近的开启定位的祖先元素的定位）

绝对定位会使元素提升一个层级；

会改变元素的性质，内联元素的宽度高度默认都被内容撑开；

（3）固定定位（相对于浏览器窗口）

与绝对定位的区别是，不受父级影响；

不随滚动条滚动；——可以做广告、到顶部、视频视窗

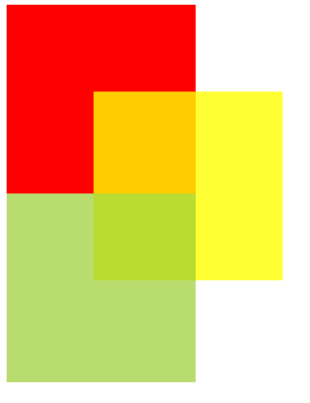
**三十五、元素层级**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <meta charset="utf-8">  <title>元素层级</title>  <style type="text/css">  .box1{  width: 200px;  height: 200px;  background-color: red;  position: relative;  }    .box2{  width: 200px;  height: 200px;  background-color: yellow;  position: absolute;  left: 100px;  top: 100px;  z-index: 1;  }    .box3{  width: 200px;  height: 200px;  background-color: yellowgreen;  /\* 层级越高，越优先显示  只有开启定位的元素可以使用该属性  父元素不能覆盖子元素\*/  z-index: 2;  position: relative;  }  </style>  </head>  <body>  <div class="box1"></div>  <div class="box2"></div>  <div class="box3"></div>  </body>  </html> |



**三十六、opacity**

可以设置0~1之间的值，0表示全透明，1表示不透明。



**三十七、背景**

背景颜色在背景图片的后面；

——重复的背景图片都可以使用平铺的方式去实现，而不需要写复杂的css样式。

background-position可以实现背景图片定位。

**三十八、雪碧图的制作和使用**

（1）制作：可以使用ps制作。

（2）使用：在css中设置不同的div，去显示雪碧图的某一个局部，需要时则加入样式。